Requested Patent: JP9319614A

Title: REMOTE TYPE DEBUGGING SYSTEM ;

Abstracted Patent: JP9319614 ;

Publication Date: 1997-12-12 ;

Inventor(s): SATO TAKAO ;

Applicant(s): NEC CORP ;

Application Number: JP19960160752 19960531 ;

Priority Number(s): ;

IPC Classification: G06F11/28 ;

Equivalents:

ABSTRACT:

PROBLEM TO BE SOLVED: To intuitively comprehend the execution conditions of a compiler type program by displaying a row number corresponding to the code address of a stop point returned from a program debugger together with a source program row. SOLUTION: When program row number information 2 storing the code address corresponding to the row number of the program is inputted, a stop point setting means 3-4 sets all the code addresses of row number information in a program debugger 4 as the stop point. Next, an application program 5 is executed by the debugger 4 and at a time point when that execution reaches the stop point, the debugger 4 returns the code address of the stop point. The returned address is inputted from a real stop point input means 3-3, the row number information is retrieved by a row number retrieving means 3-2, and the row number corresponding to the current stop point is provided. Next, this row number is displayed on a program source display means 1 by a row number instructing means 3-1 and the correspondent source program row is displayed.

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（A）　(11)特許出願公開番号
特開平9－319614
(43)公開日　平成９年(1997)12月12日

(51)Int.Cl.[6]	識別記号	庁内整理番号	ＦＩ		技術表示箇所
Ｇ０６Ｆ　11/28	３１０		Ｇ０６Ｆ　11/28	３１０Ｄ	

審査請求　有　　請求項の数３　ＦＤ　（全　６　頁）

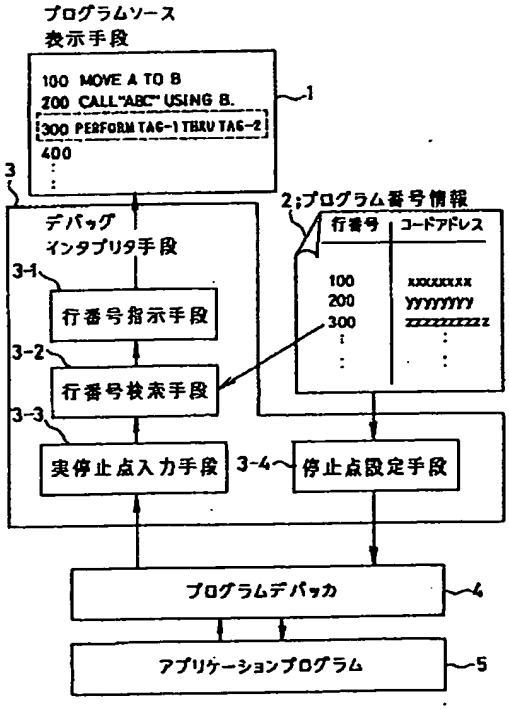
(21)出願番号　特願平8－160752

(22)出願日　平成８年(1996)５月31日

(71)出願人　000004237
日本電気株式会社
東京都港区芝五丁目７番１号
(72)発明者　佐藤　孝夫
東京都港区芝五丁目７番１号　日本電気株式会社内
(74)代理人　弁理士　加藤　朝道

(54)【発明の名称】　リモート型デバッグ方式

(57)【要約】
【課題】コンパイラ方式で生成された実行アプリケーションプログラムに対するデバッグシステムに実行行番号を逐次表示し、実行状況を直感的な理解を可能として操作性の向上を可能とするデバッグ方式の提供。
【解決手段】プログラムをソースイメージで表示するプログラムソース表示手段１と、プログラムの行番号と対応するコードアドレスを格納したプログラム行番号情報２と、停止点（ブレークポイント）の設定手段３－４、デバッガから実停止点のコードアドレスを入力する手段３－３、入力したコードアドレスから行番号を検索する手段３－２、行番号を指示する手段３－１からなるデバッグインタプリタ手段３を備え、アプリケーションプログラムの実行を行番号でトレース可能とする。



【特許請求の範囲】
【請求項１】ソースプログラムに割り付けられる行番号と該行番号に対応するコードアドレスとの情報を記憶する記憶手段を備え、
コードアドレスにより停止点が設定され実行時に停止点の情報をコードアドレスとして返却するプログラムデバッガによりプログラムを実行する際、前記プログラムデバッガから返却された停止点のコードアドレスに対応する行番号を前記記憶手段を参照して検索し該行番号を該行番号が割り付けられたソースプログラム行と共に識別可能に表示し、
コンパイラ方式のプログラム言語により記述されたプログラムについてソースプログラムイメージで実行行番号にてトレース可能としたことを特徴とするプログラムデバッグ方式。
【請求項２】アプリケーションプログラムをデバッグするデバッグ方式において、
プログラムをソースイメージで表示するプログラム表示手段と、
プログラムの行番号と対応するコードアドレスとを格納したプログラム行番号情報と、停止点設定手段、実停止点入力手段、行番号検索手段、行番号指示手段からなるデバッグインタプリタ手段と、
を備え、
前記停止点設定手段は、コードアドレスにより停止点が設定され実行時に停止点をコードアドレスとして返却するプログラムデバッガに対して、前記プログラム行番号情報の全てのコードアドレスを停止点として設定し、
前記実停止点入力手段は、前記プログラムデバッガによりアプリケーションプログラムを実行し停止点に達した時点で前記プログラムデバッガから返却された実停止点のコードアドレスを入力し、
前記行番号検索手段は、前記実停止点入力手段にて入力した前記コードアドレスから前記プログラム行番号情報を検索してログラム行番号を得、
前記行番号指示手段は、前記検索したプログラム行番号を前記プログラムソース表示手段に指示することを逐次繰り返して、前記アプリケーションプログラムの実行行番号を表示することを特徴とするプログラムアニメート方式。
【請求項３】複数の計算機システムがネットワーク接続されたシステムにおいて、少なくとも一つの計算機システムが請求項２記載の前記デバッグインタプリタ手段を備え、
前記デバッグインタプリタ手段の前記実停止点入力手段が、前記デバッグインタプリタを備えた自計算機システム又はネットワーク接続された他の計算機システムで起動される、コードアドレスにより停止点が設定され実行時に停止点をコードアドレスとして返却する、プログラムデバッガからのコードアドレスを入力し、
計算機システム固有のプログラムデバッガによるデバッグに対して、プログラム表示手段にて所定の統一仕様のグラフィカルユーザインターフェースにてプログラムの実行を行番号に基づきトレース自在としたことを特徴とするリモート型デバッグ方式。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明はプログラムのデバッグ方式に関し、特にアプリケーションプログラムの実行状況を実行ライン番号（実行行番号）として逐次表示するというプログラムアニメート方式に関する。
【０００２】
【従来の技術】プログラムデバッグの従来方式として、アプリケーションプログラムの実行状況として実行ラインを逐次表示する場合、アプリケーションプログラムをインタプリタ方式で逐次実行し、アプリケーションプログラムの該当する行（ライン）を逐次表示する方式をとっていた。
【０００３】この従来の方式では、インタプリタを持たないコンパイラ方式のプログラム言語について、アプリケーションプログラムの実行状況として実行ラインを逐次表示するというプログラムアニメート方式を実現することができない。
【０００４】
【発明が解決しようとする課題】上記したように、従来のプログラムデバッグ方式においては、アプリケーションプログラムをインタプリタ方式で逐次実行しアプリケーションプログラムの該当行を表示する方式をとるため、アプリケーションプログラムの実行状況として実行行を逐次表示するプログラムアニメート方式を実現するためには、当該アプリケーションプログラムを一文ずつ（１ラインずつ）逐次解釈実行するインタプリタシステムが必要とされていた。
【０００５】一方、プログラム言語翻訳システムは、アプリケーションプログラムの高速実行を図るため、コンパイラ方式により目的プログラムを生成し、命令を逐次解釈実行するため実行速度が遅いインタプリタ方式をとらないシステムが多数存在している。
【０００６】このため、コンパイラ方式をとるプログラム言語翻訳システムでは、プログラムの実行状況を直感的に理解することができる有効な実行ラインを逐次表示するプログラムアニメート方式を実現することはできず、プログラムの実行状況を直感的に理解することが困難とされていた。
【０００７】従って、本発明は、上記問題点に鑑みてなされたものであって、その目的は、コンパイラ方式で生成されるアプリケーションプログラムの実行状況を直感的に理解容易とし操作性を向上させるプログラムのアニメート方式を提供することにある。
【０００８】

【課題を解決するための手段】上記目的を達成するために、本発明のプログラムデバッグ方式は、ソースプログラムに割り付けられる行番号と該行番号に対応するコードアドレスとの情報を記憶する記憶手段を備え、コードアドレスにより停止点が設定され実行時に停止点の情報をコードアドレスとして返却するプログラムデバッガによりプログラムを実行する際、前記プログラムデバッガから返却された停止点のコードアドレスに対応する行番号を前記記憶手段を参照して検索し該行番号を該行番号が割り付けられたソースプログラム行と共に識別可能に表示し、コンパイラ方式のプログラム言語により記述されたプログラムについてソースプログラムイメージで実行行番号にてトレース可能としたことを特徴とする。

【０００９】また、本発明は、好ましくは、プログラムをソースイメージで表示するプログラム表示手段と、プログラムの行番号と対応するコードアドレスとを格納したプログラム行番号情報と、停止点設定手段、実停止点入力手段、行番号検索手段、行番号指示手段からなるデバッグインタプリタ手段と、を備え、前記停止点設定手段は、コードアドレスにより停止点が設定され実行時に停止点をコードアドレスとして返却するプログラムデバッガに対して、前記プログラム行番号情報の全てのコードアドレスを停止点として設定し、前記実停止点入力手段は、前記プログラムデバッガによりアプリケーションプログラムを実行し停止点に達した時点で前記プログラムデバッガから返却された実停止点のコードアドレスを入力し、前記行番号検索手段は、前記実停止点入力手段にて入力した前記コードアドレスから前記プログラム行番号情報を検索してログラム行番号を得、前記行番号指示手段は、前記検索したプログラム行番号を前記プログラムソース表示手段に指示することを逐次繰り返して、前記アプリケーションプログラムの実行行番号を表示することを特徴とするプログラムアニメート方式を提供する。

【００１０】本発明の概要を以下に説明する。本発明においては、コードアドレスによる停止点の設定と実行時に停止点をコードアドレスとして返却する一般的なプログラムデバッガに対して、プログラム行番号情報の全てのコードアドレスを停止点として設定し、プログラムデバッガによりアプリケーションプログラムを実行し停止点に達した時点でプログラムデバッカから返却された実停止点のコードアドレスを入力し、入力したコードアドレスからプログラム行番号情報を検索してプログラム行番号を得て、検索したプログラム行番号をプログラム行番号表示手段で指示することを逐次繰り返して、アプリケーションプログラムの実行行番号を表示するプログラムアニメート方式を実現し、コンパイラ方式で生成されたアプリケーションプログラムの実行状況の直感的な理解を可能として操作性を向上させる。

【００１１】このように、本発明によれば、アプリケーションプログラムの実行行番号を表示するプログラムアニメート方式を実現することにより、アプリケーションプログラムの実行行状況を、プログラムソースの形式にて、どのプログラム行が実行されてどのような順序で実行されているかを表示することができる。

【００１２】

【発明の実施の形態】本発明の実施の形態について図面を参照して以下に説明する。

【００１３】図1は、本発明の実施の形態を説明するための図である。図1を参照すると、本発明の実施の形態は、プログラムソース表示手段１と、プログラムの行番号と対応するコードアドレスを格納したプログラム行番号情報2と、行番号指示手段３－１、行番号検索手段３－2、実停止点入力手段３－３、及び停止点設定手段３－4からなるデバッグインタプリタ手段3と、を備え、さらにプログラムデバッガ４、アプリケーションプログラム５を備えて構成される。

【００１４】図2は、本発明の実施の形態の処理動作を説明するための流れ図である。

【００１５】プログラムの行番号と対応するコードアドレスを格納したプログラム行番号情報１を入力する（ステップ２１）。

【００１６】次に、コードアドレスによる停止点（ブレークポイント）の設定と実行時に停止点をコードアドレスとして返却する一般的なプログラムデバッガ４に対して、停止点設定手段３－４を用いてプログラム行番号情報の全てのコードアドレスを停止点として設定する（ステップ２２）。このステップ２２の処理によって、アプリケーションプログラムの全てソース行に対応したコードアドレスがアプリケーションプログラムの停止点として設定される。

【００１７】次に、プログラムデバッガ４によりアプリケーションプログラムを実行する（ステップ２３）。

【００１８】アプリケーションプログラムの実行が停止点に達した時点でプログラムデバッカ４は、停止点のコードアドレスを返却する（ステップ２４）。

【００１９】返却されアプリケーションプログラムの現在の停止点のコードアドレスを実停止点入力手段３－３により入力し（ステップ２５）、入力したコードアドレスから行番号検索手段３－２によりコードアドレスに対応したプログラム行番号情報を検索して現在の停止点に対応するプログラム行番号を得る（ステップ２６）。

【００２０】次に、検索したプログラム行番号を行番号指示手段３－１によりプログラムソース表示手段１に指示してプログラム行番号に対応するソースプログラム行を表示する（ステップ２７）。

【００２１】以降、アプリケーションプログラムが終了するかデバッグ作業自体が終了するまでステップ２３～２７を逐次繰り返して実行する。以上の操作によりプログラムソース表示手段１により表示されるソースプログ

ラムは、アプリケーションプログラムの実行ライン番号を逐次表示するプログラムアニメート方式を実現することが可能となる。

【0022】

【実施例】次に、本発明の一実施例を説明する。本発明の一実施例の構成は、本発明の実施の形態の説明で参照した図1に示した構成と同様とされるため説明を省略する。本発明の一実施例で利用するプログラムデバッガ4は、コードアドレス形式での停止点の設定とデバッグ実行にコードアドレス形式での停止点の返却が可能であるような一般的なプログラムデバッガを想定する。

【0023】また、本発明の実施例を説明するためのアプリケーションプログラムを図3に示すようなCOBOL言語によるプログラム例で説明する。

【0024】ここで100、200、300はプログラムの行番号を示す。

【0025】プログラムソース表示手段1は、このCOBOLプログラムをソースプログラムイメージで表示している。

【0026】また、プログラムソース表示手段1はソース行の表示方法として、ソース行番号が指定されると、そのソース行をディスプレイ上に輝度反転して表示する。すなわち図1は、プログラムソース表示手段1は300行の表示指示に対応して、300　PERFORM　TAG－1　THRU　TAG－2．を輝度反転してディスプレイ表示した状態を示している（図1ではこれを破線で囲んで示している）。

【0027】また、COBOLプログラム例に対応してプログラム行番号情報2は、図1に示すような内容であるとする。このプログラム行番号情報2は、それぞれ、
プログラム行番号100行の開始コードアドレスがXXXXXXXX
プログラム行番号200行の開始コードアドレスがYYYYYYYY
プログラム行番号300行の開始コードアドレスがZZZZZZZZZ
であることを示している。

【0028】図2の流れ図を参照して、本発明の一実施例の処理を以下に説明する。

【0029】プログラムの行番号と対応するコードアドレスを格納したプログラム行番号情報1を入力し（ステップ21）、コードアドレスによる停止点の設定と実行時に停止点をコードアドレスとして返却する一般的なプログラムデバッガ4に対して、停止点設定手段3－4を用いてプログラム行番号情報の全てのコードアドレスを停止点として設定する（ステップ22）が、例えば、
プログラム行番号100行の開始コードアドレスがXXXXXXXX
プログラム行番号200行の開始コードアドレスがYYYYYYYY
プログラム行番号300行の開始コードアドレスがZZZZZZZZZ
以下省略、の開始コードアドレスXXXXXXXX、YYYYYYYY、ZZZZZZZZZ、…をCOBOLプログラム例の停止点として全て設定する。

【0030】この操作によりプログラムデバッガ4は、図3に示したCOBOLプログラムを実行時に、実行アドレスがXXXXXXXX、YYYYYYYY、ZZZZZZZZZ、…に達する度に、一時停止して停止している実行アドレスを返却する。

【0031】次にプログラムデバッガ4によりCOBOLプログラムを実行し（ステップ23）、アプリケーションプログラムの実行が最初に停止点に達した時点でプログラムデバッガ4は、停止点のコードアドレスXXXXXXXXを返却し（ステップ24）、返却されアプリケーションプログラムの現在の停止点のコードアドレスXXXXXXXXを実停止点入力手段3－3により入力して（ステップ25）、入力したコードアドレスから行番号検索手段3－2によりコードアドレスXXXXXXXXに対応したプログラム行番号情報を検索して現在の停止点に対応するプログラム行番号100を得る（ステップ26）。次に検索したプログラム行番号100を行番号指示手段3－1によりプログラムソース表示手段1に指示してプログラム行番号に対応するソースプログラム行を表示する（ステップ27）。プログラムソース表示手段1は、予めCOBOLプログラムの全体を表示しており、ステップ27でプログラム行番号100の表示指示を受けて100行のソースプログラムを反転表示する。

【0032】次に、ステップ23、及び24の操作において、COBOLプログラムはYYYYYYYYで停止する。前回の処理と同様にステップ24、25、26の操作によりプログラム行番号200を検索して、プログラム行番号200の表示をプログラムソース表示手段1に指示する。プログラム行番号200の表示指示を受けて200行のソースプログラムを反転表示する。以降同様の操作でZZZZZZZZZ停止した場合には、300行に対応するCOBOLプログラムの行が反転表示される。

【0033】以上のように、COBOLプログラム例の実行に合わせて100行、200行、300行に対応したソースがプログラムソース表示手段1により順次輝度反転して表示されることとなり、COBOLプログラムの実行をソース形式で直感的に理解することができる。

【0034】本発明の実施例に係るプログラムデバッグ方式においては、複数の計算機システムがネットワークに接続されてなる環境において、各計算機システムで異なったデバッガを用いてプログラム開発が行われる際に、デバッガインタプリタ3を一つの計算機システムにインストールしておき、他の計算機システムにおいて起

動されるデバッガ4によりアプリケーションプログラムのデバッグ時に、停止点のコードアドレスをネットワークを介して上記一つの計算機システムにインストールされているデバッガインタプリタ3の実停止点入力手段3－3に転送することにより、プログラムソース表示手段1において、共通のグラフィカルユーザインターフェース（GUI）にてプログラムの実行ライン番号をトレース表示することが可能とされ、これにより例えばクライエント／サーバ方式のリモート型のデバッグ環境を実現できる。また、本発明の実施例では、コンパイラ方式の言語としてCOBOLを例に説明したが、本発明はその他PL／1、Fortran等に対しても同様にして適用することができる。

【0035】

【発明の効果】以上説明したように、本発明によれば、コンパイラ方式で生成されたアプリケーションプログラムに対してもプログラムアニメート方式を実現し、アプリケーションプログラムの実行状況を直感的に理解することを可能とし、その結果操作性を向上させると共に、プログラムの開発効率及び生産性を向上させることができるという効果を有する。

【図面の簡単な説明】

【図1】本発明の実施の形態を説明するための図である。

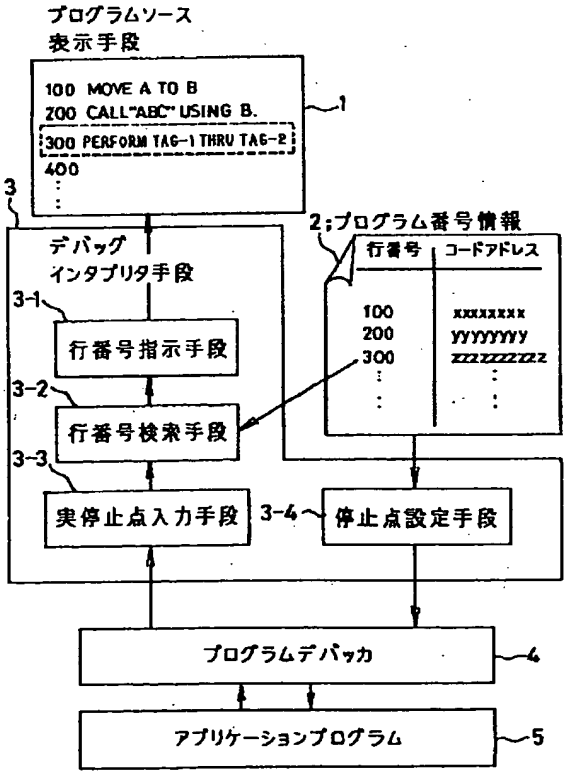
【図2】本発明の実施の形態の処理動作を説明するための流れ図である。

【図3】本発明の実施例を説明するためのアプリケーションプログラムの例を示す図である。

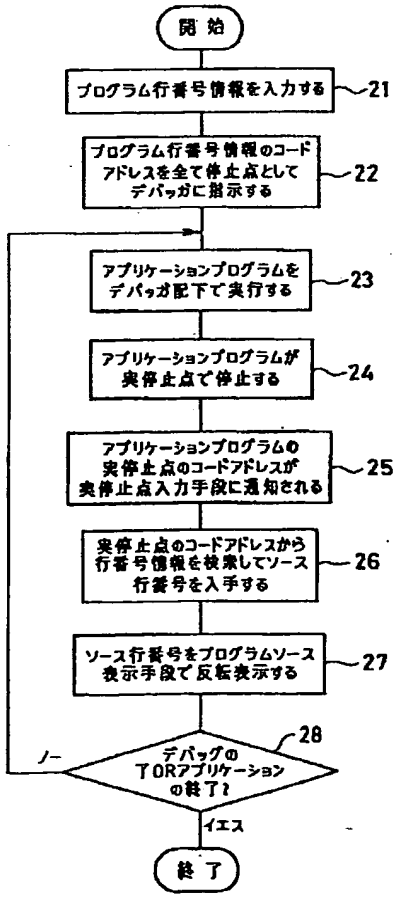
【符号の説明】

1　プログラムソース表示手段
2　プログラム行番号情報
3　デバッグインタプリタ手段
3－1　行番号指示手段
3－2　行番号検索手段
3－3　実停止点入力手段
3－4　停止点設定手段
4　プログラムデバッガ
5　アプリケーションプログラム

【図1】



【図2】



【図3】

```
100 MOVE A TO B.
200 CALL "ABC" USING B.
300 PERFORM TAG-1 THRU TAG-2.
400 ...
```